

IN THE CLAIMS

Please amend the claims as follows.

For the Examiner's convenience, a list of all claims is included below.

1-23 (Cancelled)

24. (Currently Amended) A computer system for dynamically and automatically loading and unloading a software library to and from memory in a computer, said software library including one or more library routines and capable of being referenced by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

software to perform operations comprising ~~for performing the following steps:~~

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library, said loading occurring prior to when a library routine of the software library is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

automatically unloading from memory the library implementation module after the execution of the library routine is completed, wherein the software library is scheduled to be unloaded automatically without an explicit unload request from the application or other software module.

25. (Currently Amended) The computer system as defined in Claim 24 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module and wherein the software library is scheduled to be removed before the execution of the library routine is completed.

26. (Previously Presented) The computer system as defined in Claim 24 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module.

27. (Currently Amended) A computer system for dynamically and automatically loading and unloading a software library to and from memory in a computer, said computer library including one or more library routines and capable of being referenced by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

software to perform operations comprising ~~for performing the following steps:~~

loading into memory a library implementation module for a software library, said loading occurring prior to when a library routine of the software library is to be executed by the application or other software module, said loading comprising ~~the steps of:~~

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory and scheduled to be unloaded from memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

automatically scheduling the unloading of the library implementation module; and

unless the scheduled unload has been cancelled, automatically unloading the library implementation module as scheduled, wherein the software library is scheduled for unload and unloaded automatically without an explicit unload request from the application or other software module.

28. (Original) The computer system as defined in Claim 27 wherein said loading of a library loader for each potentially needed library occurs either just prior to or contemporaneous with the execution of the application or other software module.

29. (Previously Presented) The computer system as defined in Claim 27 wherein said loading of a library loader for each potentially needed library occurs after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module.

30. (Currently Amended) The computer system as defined in Claim 27 wherein said ~~step of~~ scheduling the unloading of the library implementation module includes delaying the unloading of the library for a specified delay and unloading the library implementation module upon completion of the delay.

31. (Previously Presented) The computer system as defined in Claim 30 wherein said specified delay is a time delay.
32. (Original) The computer system as defined in Claim 31 wherein time delay is fixed.
33. (Original) The computer system as defined in Claim 31 wherein said time delay is variable.
34. (Previously Presented) The computer system as defined in Claim 30 wherein said specified delay is based on system resources.
35. (Currently Amended) The computer system as defined in Claim 27 wherein said ~~step of~~ scheduling the unloading of the library implementation module comprises ~~comprises the steps of:~~
- setting an associated criteria;
 - unsetting an associated criteria of said scheduled unloading is cancelled;
 - and
 - resetting the associated criteria of the library implementation module is again scheduled for unloading.
36. (Currently Amended) ~~[[The]]~~ A computer system for dynamically and automatically loading and unloading a software library to and from memory in a

computer, said software library capable of being referenced by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

software to perform operations comprising ~~for performing the following steps:~~

loading a software library into memory; and

automatically unloading said software library from memory after completion of execution of said library routine, wherein the software library is scheduled to be unloaded automatically without an explicit unload request from the application or other software module.

37. (Currently Amended) The computer system of Claim 36 wherein said unloading of said software library comprises ~~the steps of:~~

scheduling the unloading of the library implementation module;

setting an associated criteria;

checking whether said set associated criteria has been met; and

unloading the library implementation module if after said associated criteria has been met, said library implementation module is not being referenced by an application or other software module.

38-51 (Cancelled)

52. (Currently Amended) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software to perform operations comprising
~~for performing the following steps:~~

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising ~~the steps of:~~

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory and
scheduled to be unloaded from memory, then canceling the scheduled unloading
of the library implementation module;

executing the loaded library implementation module for the
library routine being executed; and

scheduling the unloading of the library implementation module,
wherein said ~~step of~~ scheduling the unloading of the library implementation
module includes setting a time delay and unloading the library implementation
module if after said time delay has been meet, said library implementation
module is into in use.

53. (Previously Presented) The computer system as defined in Claim 52
wherein said time delay is fixed.

54. (Previously Presented) The computer system as defined in Claim 52
wherein said time delay is variable.

55. (Currently Amended) A computer system for dynamically loading a
software library into memory in a computer, said software capable of being
used by an application or other software module, said computer system
comprising:

one or more library structures, each library structure corresponding to a
software library, each library structure including a library implementation
module containing code for implementing the corresponding software library

and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software to perform operations comprising
~~for performing the following steps:~~

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising ~~the steps of:~~

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory and
scheduled to be unloaded from memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said ~~step of~~ scheduling the unloading of the library implementation module includes setting an associated criteria based on system resources and unloading the library implementation module if after said associated criteria has been met, said library implementation module is not in use.

56. (Currently Amended) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software to perform operations comprising
~~for performing the following steps:~~

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising ~~the steps of:~~

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory and
scheduled to be unloaded from memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said ~~step of~~ scheduling the unloading of the library implementation module includes setting an associated criteria, unsetting an associated criteria if said scheduled unloading is cancelled, and resetting the associated criteria if the library implementation modules is again scheduled for unloading.

57. (Currently Amended) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software to perform operations comprising ~~for performing the following steps:~~

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading comprising ~~the steps of:~~

determining whether the library implementation module is in memory;

if the library implementation module is not in memory, then loading the library implementation module; and

if the library implementation module is in memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said ~~step of~~ scheduling the unloading of the library implementation module includes setting an associated criteria, unsetting an associated criteria if said scheduled unloading is cancelled, and resetting the associated criteria if the library implementation modules is again ascheduled for unloading.

58. (Currently Amended) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software to perform operations comprising
~~for performing the following steps:~~

loading a software library into memory just prior to or
contemporaneous with execution of a library routine in said software library;
and

unloading said software library from memory after completion of
execution of said library routine, wherein said unloading of said software library
includes scheduling the unloading of the library implementation module, setting
an associated criteria, checking whether said set associated criteria has been
met, and unloading the library implementation module if after said associated
criteria as been met, said library implementation module is not in use.

59. (Currently Amended) A computer-readable medium having stored
thereon instructions for causing a computer to perform operations comprising
~~the following steps:~~

determining one or more software libraries which are potentially needed
during execution of an application or other software module;

loading into memory a library loader for each potentially needed library,
said loading occurring after execution of the application or other software
module begins but before any library routine of the potentially needed library is
executed by the application or other software module;

loading into memory a library implementation module for a software
library containing a library routine, said loading occurring just prior to when

said library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

automatically unloading from memory the library implementation module after the execution of the library routine is completed, wherein the library implementation module is scheduled to be unloaded from memory.

60. (Currently Amended) A computer-readable medium having stored thereon instructions for causing a computer to perform operations comprising the following steps:

determining one or more software libraries which are potentially needed during execution of an application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring anytime before any library of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said loading occurring just prior to when said library routine is to be executed by the application or other software module, said loading including determining whether the library implementation modules is in memory, if the library implementation module is not in memory, then loading the library implementation module, and if the library implementation module is in memory and scheduled to be unloaded from

memory, then canceling the scheduled unloading of the library implementation module;

executing the loaded library implementation module for the library routine being executed; and

scheduling the unloading of the library implementation module, wherein said ~~step of~~ scheduling the unloading of the library implementation module includes setting a time delay and unloading the library implementation module if after said time delay has been met, said library implementation module is not in use.

61. (Previously Presented) The computer-readable medium as defined in Claim 60 wherein said time delay is fixed.

62. (Previously Presented) The computer-readable medium as defined in Claim 60 wherein said time delay is variable.

63. (Currently Amended) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an application or other software module, said computer system comprising:

one or more libraries, each software library containing one or more library routines, each software library having a library loader and a library implementation module, the library loader specifying entry points

corresponding to entry points in the software library and the library implementation module containing computer code to implement the software library, the library loader handling ~~the step of~~ unloading the software library;

means for loading into memory a library loader for each potentially needed library, a library loader for a library being loaded by the time the application or other software module executes a routine in that library;

means for loading into memory a library implementation module for a software library, said loading occurring prior to when a library routine of the software library is executed by the application or other software module, said loading handled by the library loader for the software library;

means for determining whether any application or other software module is referencing the software library; and

means for automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module, wherein the software library is scheduled to be unloaded automatically without an explicit unload request from the application or other software module.

64. (Currently Amended) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an application or other software module, said computer system comprising:

one or more libraries, each software library containing one or more library routines

means for determining whether any application or other software module is referencing the software library; and

means for automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module, wherein the software library is scheduled to be unloaded automatically without an explicit unload request from the application or other software module, wherein said means for automatically unloading the software library comprises means for delaying the unloading of the software library for a specified delay and means for unloading the software library upon completion of the delay.

65. (Previously Presented) The computer system as defined in Claim 64, wherein said means for automatically unloading the software library further comprises:

means for determining at completion of the delay whether the software library is being referenced by an application or other software module; and

means for unloading the software module only if the software library is not being referenced by an application or other software module;

66. (Previously Presented) The computer system as defined in Claim 64 further comprising:

when an application or other software module references or otherwise invokes a routine in a software library, means for determining whether the software library is subject to a delayed unloading; and

if the software library is subject to a delayed unloading means for canceling the delayed unloading.

67. (Previously Presented) The computer system as defined in Claim 64 wherein said specified delay is a time-based delay.

68. (Currently Amended) A computer system for automatic unloading of a dynamically-loaded software library from memory in a computer, said software library including one or more library routines and capable of being referenced or otherwise invoked by an application or other software module, said computer system comprising:

a processor;

a memory;

a disk;

one or more software libraries stored on said disk, each software library containing one or more library routines;

software operated on by said processor to perform operations

comprising the following steps:

determining whether any application or other software module is referencing the software library; and

automatically unloading the software library from memory if it is determined that the software library is not being referenced by any application or other software module, wherein the software library is scheduled to be unloaded automatically without an explicit unload request from the application or other software module.

69. (Currently Amended) The computer system as defined in Claim 68 wherein said software library is comprised of a library loader and a library implementation module, the library loader specifying entry points corresponding to entry points in the software library and the library implementation module containing computer code to implement the software library, the library loader handling ~~the step of~~ unloading the software library, said software performing operations comprising the following steps:

loading into memory a library loader for each potentially needed library, a library loader for a library being loaded by the time the application or other software module executes a routine in that library; and

loading into memory a library implementation module for a software library, said loading occurring just prior to when a library routine of the software library is executed by the application or other software module, said loading handled by the library loader for the software library.

70. (Currently Amended) The computer system as defined in claim 68 wherein said ~~step of~~ automatically unloading the software library comprises ~~the~~

~~steps of~~ delaying the unloading of the software library for a specified delay and unloading the software library upon completion of the delay.

71. (Currently Amended) The computer system as defined in claim 68, said ~~step of~~ automatically unloading the software library further comprises ~~the steps of~~:

determining at completion of the delay whether the software library is being referenced by an application or other software module; and

unloading the software module only if the software library is not being referenced by an application or other software module;

72. (Currently Amended) The computer system of claim 70 further comprising ~~the steps of~~:

when an applicatin or other software module references or otherwise invokes a routine in a software library, determining whether the software library is subject to a delayed unloading; and

if the software library is subject to a delayed unloading cnacelling the delayed unloading.

73. (Previously Presented) The computer system as defined in Claim 70 wherein said specified delay is a time-based delay.

74. (Currently Amended) A computer-readable medium having stored thereon instructions for causing a computer to perform operations comprising the following steps:

loading a software library into memory; and
automatically unloading said software library from memory after completion of execution of said library routine by delaying the unloading for a delay period and canceling a delayed unloading of a software library if a library routine of the software library is referenced by an application or other software module during the delay period.

75. (Currently Amended) A computer system for dynamically loading a software library into memory in a computer, said software capable of being used by an application or other software module, said computer system comprising:

one or more library structures, each library structure corresponding to a software library, each library structure including a library implementation module containing code for implementing the corresponding software library and a library loader containing entry points corresponding to entry points of the corresponding software library and code for loading and unloading the corresponding library implementation module; and

an operating system or other software operations comprising for performing the following steps:

determining which libraries are potentially needed during execution of the application or other software module;

loading into memory a library loader for each potentially needed library, said loading occurring after execution of the application or other software module begins but before any library routine of the potentially needed library is executed by the application or other software module;

loading into memory a library implementation module for a software library containing a library routine, said for a software library containing a library routine is to be executed by the application or other software module;

executing the loaded library implementation module for the library routine being executed; and

unloading from memory the library implementation module after the execution of the library routine is completed, wherein the library implementation module is scheduled to be unloaded automatically.